

PENERAPAN SENTIMENT ANALYSIS PADA HASIL EVALUASI DOSEN DENGAN METODE SUPPORT VECTOR MACHINE

Valonia Inge Santoso¹, Gloria Virginia², Yuan Lukito³

Program Studi Teknik Informatika
Fakultas Teknologi Informasi
Universitas Kristen Duta Wacana

valoniainge@ti.ukdw.ac.id¹, virginia@staff.ukdw.ac.id², yuan@staff.ukdw.ac.id³

Abstract - The quality of lectures can be determined by some feedbacks from students. From the feedbacks, we can give appreciations for those lectures who get good feedback from students, and evaluations for those who get bad feedback. The problem is classifying large size of feedbacks manually isn't effective and took a lot of time. Therefore, we need a system that can classify feedbacks automatically. These feedbacks will be classified into positive, negative, and neutral, usually called as sentiment analysis. Sentiment analysis implementation can be done by several methods, one of them that has a good accuracy is Support Vector Machine (SVM). SVM performance in this research is measured with the level of accuracy. The number of accuracy indicate the success level of system. The conclusion of this research is factors that affects the accuracy. The factors are the range of each classes and number of unique words in the training document.

Keywords: sentiment analysis, support vector machine, inverse matrix

I. PENDAHULUAN

Universitas Kristen Duta Wacana menerapkan kuesioner online dalam rangka penilaian terhadap dosen. Kuesioner ini diisi oleh mahasiswa dan berisi opini positif, negatif, atau netral. Saat ini, untuk perekapan dan klasifikasi hasil kuesioner untuk evaluasi dosen ini dilakukan secara manual. Pengklasifikasian secara manual memang menghasilkan data yang akurat karena manusia dapat membedakan dengan tepat apakah kata atau kalimat tersebut bermakna positif, negatif, atau netral, namun hal ini tidak efektif dan tidak menutup kemungkinan adanya kalimat ambigu yang sulit diklasifikasi walaupun oleh manusia sekalipun. Selain itu, pengklasifikasian secara manual membutuhkan banyak waktu dan tenaga. Untuk mengatasi masalah ini, dibutuhkan sistem yang dapat mengklasifikasikan opini ini ke dalam kelas sentimen positif, negatif, atau netral secara otomatis. Klasifikasi ke tiga kelas sentimen ini disebut *sentiment analysis*.

Sistem ini menggunakan metode *support vector machine* (SVM) karena pada penelitian – penelitian sebelumnya yang pernah dilakukan, metode ini memiliki nilai akurasi yang cukup tinggi. Dengan diterapkannya metode SVM, sistem ini diharapkan dapat melakukan *sentiment analysis* dengan cepat, mudah, dan dengan tingkat akurasi serta efektivitas yang cukup tinggi.

II. LANDASAN TEORI

2.1 Text Mining dan Sentiment Analysis

Text mining mengacu pada proses mengambil informasi berkualitas tinggi dari teks. Informasi yang diambil biasanya mengacu ke beberapa kombinasi relevansi, kebaruan, dan *interestingness* (Saraswati, 2011).

Sentiment analysis atau yang biasa disebut dengan *Opinion mining* adalah riset komputasional dari opini, sentimen, dan emosi yang dituangkan secara tekstual lalu diklasifikasikan menjadi kelompok sentimen positif dan negatif (Feizar, Indrianti, & Yudistira, 2014). Secara umum, *sentiment analysis* dibagi menjadi 2 kategori besar (Wicaksono, 2011), yaitu:

1. *Coarse – grained sentiment analysis*: Proses analisis dan klasifikasi orientasi sebuah dokumen secara keseluruhan. Orientasi ini dibagi menjadi 3 jenis yaitu positif, netral, dan negatif, akan tetapi ada juga yang menjadikan nilai orientasi bersifat kontinu / tidak diskrit.
2. *Fined – grained sentiment analysis*: Obyek yang diklasifikasi tidak pada level dokumen melainkan pada level kalimat dalam sebuah dokumen. Sebagai contoh:
 - a. Cara pengajaran kurang menarik → negatif
 - b. Kuliah terlaksana dengan tepat waktu → positif*Sentiment analysis* terdiri dari 3 subproses (Wicaksono, 2011) yaitu:
 1. *Subjectivity Classification*: menentukan kalimat yang merupakan opini.
 2. *Orientation Detection*: Pengklasifikasian opini ke dalam kelas positif, negatif, atau netral.
 3. *Opinion Holder and Target Detection*: menentukan bagian yang merupakan *opinion holder* (pemberi opini) dan bagian yang merupakan target.

2.2 Pembobotan TF-IDF

Bobot TF-IDF merupakan bobot setiap kata pada setiap dokumen. Untuk memperoleh nilai TF – IDF menggunakan Persamaan 2. Untuk mendapatkan nilai TF – IDF dibutuhkan nilai idf_t . Nilai idf_t dapat dicari dengan Persamaan 1.

$$idf_t = \log_{10} (N/df_t) \quad (1)$$

Dimana df_t adalah jumlah dokumen yang mengandung suatu *term* dan N adalah total dokumen yang diuji.

$$TF - IDF = tf \times idf_t \quad (2)$$

idf adalah jumlah dokumen yang berisi istilah tersebut. Diambil log dari idf untuk memberikan beberapa penghalusan. Dalam hal ini, setiap dokumen dianggap sebagai vektor dengan 1 komponen yang sesuai dengan setiap *term* yang ada di kamus beserta dengan bobot dari setiap komponen. Untuk *term* yang tidak muncul di dokumen, maka bobotnya 0.

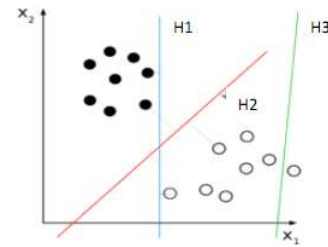
Setelah didapatkan bobot TF-IDF pada setiap *term*, maka dilakukan normalisasi pada bobot TF-IDF tersebut. Normalisasi *cosine* dilakukan dengan perhitungan yang ditunjukkan pada Persamaan 3 (Croft, Metzler, & Strohan, 2015).

$$tf - idf_{(normalisasi)} = \frac{tf-idf_{(asli)}}{\sqrt{\sum_{k=1}^t (tf-idf_{(k)})^2}} \quad (3)$$

2.3 Support Vector Machine

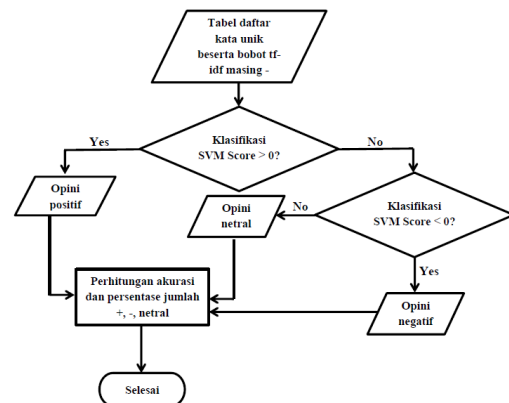
SVM merupakan salah satu metode klasifikasi dengan menggunakan metode *machine learning* (*supervised learning*) yang memprediksi kelas berdasarkan pola dari hasil proses *training* yang diciptakan oleh Vladimir Vapnik. Klasifikasi dilakukan dengan garis pembatas (*hyperlane*) yang memisahkan antara kelas opini positif dan opini negatif. Secara intuitif, suatu garis pembatas yang baik adalah yang memiliki jarak terbesar ke titik data pelatihan terdekat dari setiap kelas, karena pada umumnya semakin besar margin, semakin rendah *error* generalisasi dari pemilah. Margin adalah jarak dari suatu titik vektor di suatu kelas terhadap *hyperplane*.

Berdasarkan Gambar 1, dapat dilihat bahwa garis pembatas terbaik adalah garis H2 karena memiliki maksimum margin dan membagi menjadi 2 kelas. Sedangkan H3 tidak membagi menjadi 2 kelas, dan H1 memang membagi menjadi 2 kelas, namun margin yang dimiliki sangat kecil. *Hyperplane* yang mewakili pemisahan atau margin terbesar antara dua kelas sehingga jarak titik data terdekat di setiap sisi dimaksimalkan disebut *hyperplane* margin maksimum, dan *linier classifier* yang didefinisikannya dikenal sebagai pengklasifikasi margin maksimal. Maksimum – margin *hyperplane* dan margin untuk SVM dilatih dengan sampel dari 2 kelas yang disebut *support vector* (Saraswati, 2011).



Gambar 1. Contoh beberapa hyperlane (Saraswati, 2011)

Training pada klasifikasi SVM akan menghasilkan sebuah nilai atau pola yang akan digunakan pada proses *testing* yang bertujuan untuk pemberian label sentimen (Novantirani, Sabariah, & Effendy, n.d). Penilaian kemudian dibuat dengan menilai *score* yang merepresentasikan di sisi mana dokumen itu berada (Saraswati, 2011). Proses pengambilan keputusan dengan SVM beserta analisis berupa tingkat akurasi dan jumlah dokumen di setiap *class* positif, negatif, dan netral digambarkan seperti pada Gambar 2.



Gambar 2. Flowchart Klasifikasi dengan SVM dan Analisis

2.4 Evaluasi dan Validasi

Evaluasi performansi dilakukan untuk menguji hasil klasifikasi dengan mengukur nilai kebenaran dari sistem. Parameter yang digunakan untuk mengukur nilai kebenaran yaitu akurasi. Akurasi adalah persentase dokumen yang berhasil diklasifikasikan dengan tepat oleh sistem. Akurasi diperoleh dari hasil perhitungan yang ditunjukkan pada Persamaan 4.

$$A = \frac{TP+TN}{(TP+TN+FP+FN)} \quad (4)$$

Semua parameter untuk mendapatkan akurasi didapatkan dari *coincidence matrix* yang digambarkan seperti Gambar 3.

	Relevant	Nonrelevant
Retrieved	true positives (tp)	false positives (fp)
Not retrieved	false negatives (fn)	true negatives (tn)

Gambar 3. Coincidence Matrix
(Manning, Raghavan, & Schutze, 2009)

Proses validasi data latih berguna untuk meningkatkan nilai akurasi sistem. Proses validasi ini diharapkan akan menghasilkan batasan yang tepat untuk membagi tiga kelas sentiment secara adil dan memberikan pembobotan yang tepat sehingga akurasi dapat meningkat.

2.5 Text Preprocessing

Text preprocessing merupakan tahap awal dari *text mining*. Tujuan dari *text preprocessing* adalah untuk mempersiapkan dokumen teks yang tidak terstruktur menjadi data terstruktur yang siap digunakan untuk proses selanjutnya dengan cara menghilangkan *noise*, menyeragamkan bentuk kata dan mengurangi volume kata (Putranti & Winarko, 2014). Tahapan *text preprocessing* yang digunakan pada penelitian ini antara lain tokenisasi, *case folding*, *filtering*, *stopword removal*.

Tokenisasi digunakan untuk mendapatkan semua *term* unik yang ada dalam *sample data* sekaligus berusaha mengubah data input yang tidak menggunakan bahasa baku (misalnya: singkatan dan bahasa asing yang sering muncul) menjadi bahasa Indonesia yang baku dan sesuai dengan KBBI dan EYD. Proses membakukan sebuah kata dilakukan dengan mendaftarkan kata – kata tersebut ke dalam sebuah tabel. Tabel ini memiliki *field* kata tidak baku dan kata baku. Contoh bentuk tabel disediakan pada Tabel 1.

Tabel 1.

Tabel Kata Tidak Baku dan Kata Baku

Kata Tidak Baku	Kata Baku
Tdk	Tidak
Enggak	Tidak
Good	Baik
Tlt	Terlambat
Krg	Kurang

Case folding digunakan untuk mengubah semua karakter dokumen hasil tokenisasi menjadi *lowercase* agar dapat diurutkan secara alfabetik dan diproses selanjutnya.

Oleh karena sistem ini berpanduan klasifikasi adalah KBBI dan EYD, maka perlu adanya penyaringan agar yang diambil hanya *term* berbahasa Indonesia yang baku saja. Untuk kata yang disingkat namun sering muncul akan berusaha dibakukan. Sedangkan kata yang menggunakan bahasa asing, namun sering muncul, akan diterjemahkan ke dalam bahasa Indonesia. Tahap ini diharapkan dapat membuat kinerja sistem semakin akurat dan tepat sasaran.

Jumlah *term* yang terlalu banyak tidak efektif karena membutuhkan *memory* yang besar dan waktu pengerjaan

yang cukup lama, sehingga perlu adanya pemilahan *term*. *Term* yang spesifik dan sesuai dengan topik dokumen (dalam hal ini kasus evaluasi dosen) dapat membantu kinerja sistem agar lebih efektif dan efisien.

III. HASIL DAN PEMBAHASAN

Pada pengujian pertama, Penulis menggunakan data latih sebanyak 307 dokumen. Dokumen latih terdiri dari 103 dokumen bersentimen positif, 163 bersentimen negatif, dan 41 dokumen bersentimen netral. Pada pengujian ini akan ditentukan *range* yang tepat untuk membagi menjadi 3 kelas sentimen. Melalui pengujian ini didapatkan akurasi tertinggi adalah 49,54 %. Akurasi ini didapatkan pada *range* :

Positif: > 0 Netral: 0 Negatif: < 0

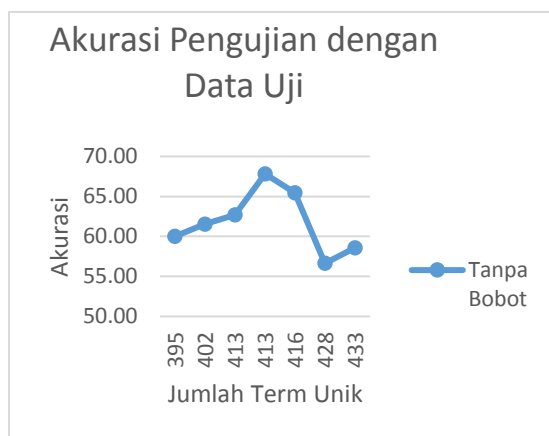
Untuk meningkatkan akurasi, Penulis mencoba untuk mengubah komposisi data latih dengan metode *k-fold cross validation* dengan $k = 7$. Melalui pengujian ini, didapatkan terjadinya perubahan tingkat akurasi yang cukup signifikan di beberapa dataset pelatihan. Hal ini membuktikan bahwa komposisi data latih turut mempengaruhi tingkat akurasi. Akurasi tertinggi didapatkan pada dataset dengan $k = 6$ dan akurasi yang dicapai adalah 67,83%. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2.

Tabel Hasil Pengujian Penerapan *K-fold cross validation*

K	Jumlah term unik	Confusion Matriks									Akurasi
		P,P	P, N	P, N t	N, P	N, N	N, N t	Nt, P	Nt, N	Nt, Nt	
1	433	24	14	0	29	40	1	3	6	1	58,56
2	416	24	13	3	21	45	1	3	6	3	65,45
3	395	23	20	1	31	39	0	1	5	7	60,00
4	402	23	18	3	21	45	3	4	4	4	61,54
5	413	27	15	1	23	44	1	7	4	3	62,71
6	413	34	6	1	23	43	2	6	8	1	67,83
7	428	24	14	0	29	40	1	5	8	0	56,64

Peningkatan akurasi berdasarkan jumlah *term* unik ditunjukkan pada Gambar 4. Melalui Gambar 4, dapat disimpulkan bahwa semakin banyak jumlah *term* unik, memiliki kecenderungan untuk meningkatkan akurasi.



Gambar 4. Grafik Akurasi Hasil Pengujian

Sistem ini memiliki nilai akurasi yang tidak terlalu tinggi dikarenakan faktor dari dokumen uji yang dimasukkan. Bila dokumen uji yang dimasukkan semua *term*-nya adalah *stopwords* maka sistem tidak dapat melakukan *sentiment analysis* pada dokumen uji tersebut. Hal ini dikarenakan sistem menerapkan *preprocessing stopwords removal* yang artinya sistem akan membuang semua *term* yang tergolong *stopwords*. Akibatnya, dokumen uji tersebut dianggap tidak mengandung *term* apapun.

Faktor lain yang menghambat kinerja sistem adalah adanya dokumen uji yang semua *term*-nya tidak terdapat di dokumen latih, dengan kata lain sistem tidak pernah mengenal *term* ini sebelumnya. Apabila ada *term* baru dari dokumen uji, perhitungan tidak dapat dilanjutkan karena *term* tersebut tidak memiliki nilai untuk dimasukan ke persamaan *hyperplane*. Alasan inilah yang membuat dokumen uji bersentimen netral sulit terdeteksi sebagai kelas sentimen netral, karena pada umumnya kalimat bersentimen netral memiliki variasi *term* yang lebih beragam daripada kelas sentimen yang lain. Tidak seperti pada kelas sentimen positif dan negatif, kelas sentimen netral cenderung tidak memiliki *term* yang khas, sehingga sulit dikenali.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil analisis penelitian, maka dapat ditarik kesimpulan bahwa sistem sudah dapat melakukan *sentiment analysis* dengan metode SVM terhadap hasil evaluasi dosen FTI UKDW Program Studi Teknik Informatika tahun ajaran 2014/2015 semester gasal. Dengan metode SVM, sistem dapat melakukan *sentiment analysis* dengan menggunakan 3 kelas sentimen. Akurasi tertinggi SVM pada sistem ini yaitu 67,83%. Akurasi tertinggi dicapai pada sistem yang tidak menerapkan perubahan bobot pada dokumen uji dan menggunakan *range* > 0 untuk kelas sentimen positif, < 0 untuk kelas sentimen negatif, dan

0 untuk kelas sentimen netral, serta pelatihan menggunakan set data latih ke 6.

Sistem pengklasifikasian ini sangat memungkinkan untuk dilakukan pengembangan lebih lanjut sesuai kebutuhan yang terus bertambah, sehingga dapat meningkatkan akurasi sistem. Saran yang diajukan Penulis dalam pengembangan sistem kedepannya adalah sebagai berikut:

1. Memperkaya variasi *term* untuk proses *filtering* yang mengatasi pembakuan kata dan menambah maupun mengurangi daftar *stopword* agar tidak terjadi terhapusnya kata yang merupakan ciri khas dari suatu *class* sentimen.
2. Menggunakan metode yang lebih baik untuk sistem penyelesaian persamaan linear agar terbentuk *hyperplane* yang lebih akurat. Akan lebih baik bila menggunakan algoritma yang dapat membentuk matriks *inverse* pada semua matriks (tidak hanya matriks persegi) agar sistem dapat mengatasi kasus jumlah dokumen latih lebih banyak dari jumlah *term* unik pada dokumen latih.
3. Pada penelitian ini, proses *sentiment analysis* dilakukan hanya berdasarkan pembobotan yang diukur dari angka kemunculan dan mengabaikan makna kata. Akan lebih baik bila pada penelitian selanjutnya diterapkan juga kedekatan suatu kata dengan kata yang lain (*semantic*). Misalnya mengkolaborasi tabel sinonim dengan tabel wordnet.
4. Berdasarkan hasil analisis, didapatkan hasil bahwa *range* pembatas antar kelas sentiment bisa saja berubah. Akan lebih baik bila sistem memiliki fitur yang memperbolehkan pengguna mengatur sendiri *range* tersebut.

DAFTAR PUSTAKA

- [1] Croft, W. B., Metzler, D., & Strohman, T. (2015). *Search Engines Information Retrieval in Practice*. Pearson Education, Inc.
- [2] Feizar, F. H., Indrianti, & Yulistira, N. (2014). *Analisis Sentimen Opini Film Berbahasa Indonesia Berbasis Kamus Menggunakan Metode Neighbor-Weighted K-Nearest Neighbor*. Universitas Brawijaya, Teknik Informatika. Malang: Universitas Brawijaya.
- [3] Manning, C. D., Raghavan, P., & Schutze, H. (2009). *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press.
- [4] Novantirani, A., Sabariah, M. K., & Effendy, V. (n.d). *Analisis Sentimen pada Twitter untuk Mengenai Penggunaan Transportasi Umum Darat Dalam Kota dengan Metode Support Vector Machine*. Universitas Telkom, Teknik Informatika. Bandung: Universitas Telkom.
- [5] Putranti, N. D., & Winarko, E. (2014, January 15). Analisis Sentimen Twitter untuk Teks Berbahasa Indonesia dengan Maximum Entropy dan Support Vector Machine. *IJCCS*, 8, 91-100.

- [6] Saraswati, N. W. (2011). *Text Mining dengan Metode Naive Bayes Classifier dan Support Vector Machines untuk Sentiment Analysis*. Universitas UDAYANA, Teknik Elektro. Denpasar: Universitas UDAYANA.
- [7] Wicaksono, A. F. (2011, Januari 20). *My Live Journals*. Dipetik September 15, 2015, dari Apa itu Sentiment Analysis / Opinion Mining?: <http://alfan-farizki.blogspot.co.id/2011/01/apa-itu-sentiment-analysis-opinion.html>